MAPLD Paper (Draft)

Scientific Computations on a NASA Reconfigurable Hypercomputer

Olaf O. Storaasli and ***Robert C. Singleterry***
Langley Research Center, Hampton Virginia

Samuel Brown
Star Bridge Systems, Midvale Utah

DRAFT (In preparation)

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

November 2002

Scientific Computations on a NAS A Reconfigurable Hypercomputer

Olaf O. Storaasli, **Robert C. Singleterry**, NASA Langley, Hampton VA
and **Samuel Brown**, Star Bridge Systems, Midvale UT

Abstract

This paper describes initial attempts to solve several pathfinder scientific applications on Reconfigurable Hypercomputers based on Field-Programmmable Gate Arrays (FPGAs). These initial applications were successful and demonstrate the feasibility of performing vector and matrix operations (dot product), transcendental (sin, log, exp, cosh…) computations , integration, differentiation, dynamic analysis and the solution of matrix equations rapidly on Langley's FPGA-based reconfigurable hypercomputer. The paper reviews FPGA capabilities and compares their advantages (speed flexibility) and over those of traditional CPUs.

Introduction

The purpose of this paper is to describe, based on first-hand research, the capabilities and limitations of Langley's Reconfigurable Hypercomputers to perform initial "pathfinder" scientific calculations involving floating-point arithmetic. Unlike most FPGA applications with no floating point calculations (image processing), NASA algorithms developed range from calculating transcendental functions to solving simultaneous matrix equations used in structural and electromagnetic analysis. The NASA reconfigurable computer is a Star Bridge Systems Hyper Algorithmic Logic (HAL)-15, the first of a family of Star Bridge Hypercomputers using Field Programmable Gate Arrays (**FPGAs**) instead of traditional CPUs, programmed via graphical icons. The traditional hardware, software and solution components are merged together in "gateware" which configures the FPGA gates for NASA applications. NASA Langley attempts to exploit the latest FPGA capability as the impact of new hardware and software has a significant impact on application performance.

NASA's pivotal research has been a catalyst in supercomputer innovations: ILLIAC (Ames-1972), Finite Element Machine (Langley-1982), MPP (Goddard-1983), Cray (Langley-1989), Intel (Delta-1992) and IBM (LaRC-1995). NASA applications were often the first to exploit/test supercomputer capability, assisting vendors to achieve stable products. NASA is exploring "reconfigurable computing" which some consider the next supercomputer breakthough as they offer reduced size, cost, power consumption and computation time. They replace fixed, sequential CPU circuitry (< 1% active each cycle) with reconfigurable, inherently parallel silicon gates (often > 80% active each cycle)[1].

Hypercomputer Hardware

At the heart of current Reconfigurable Computers are one or more FPGAs mounted on a circuit board. The NASA HAL-15 Hypercomputers include a circuit board with ten FPGAs (Fig 1) - one FPGA is located on the reverse side. Each FPGA contains 62,000

hardware gates configured as 2043 Computationl Logic Blocks (CLBs) which can be coded both in parallel on one chip and using mutiple FPGAs on one board.

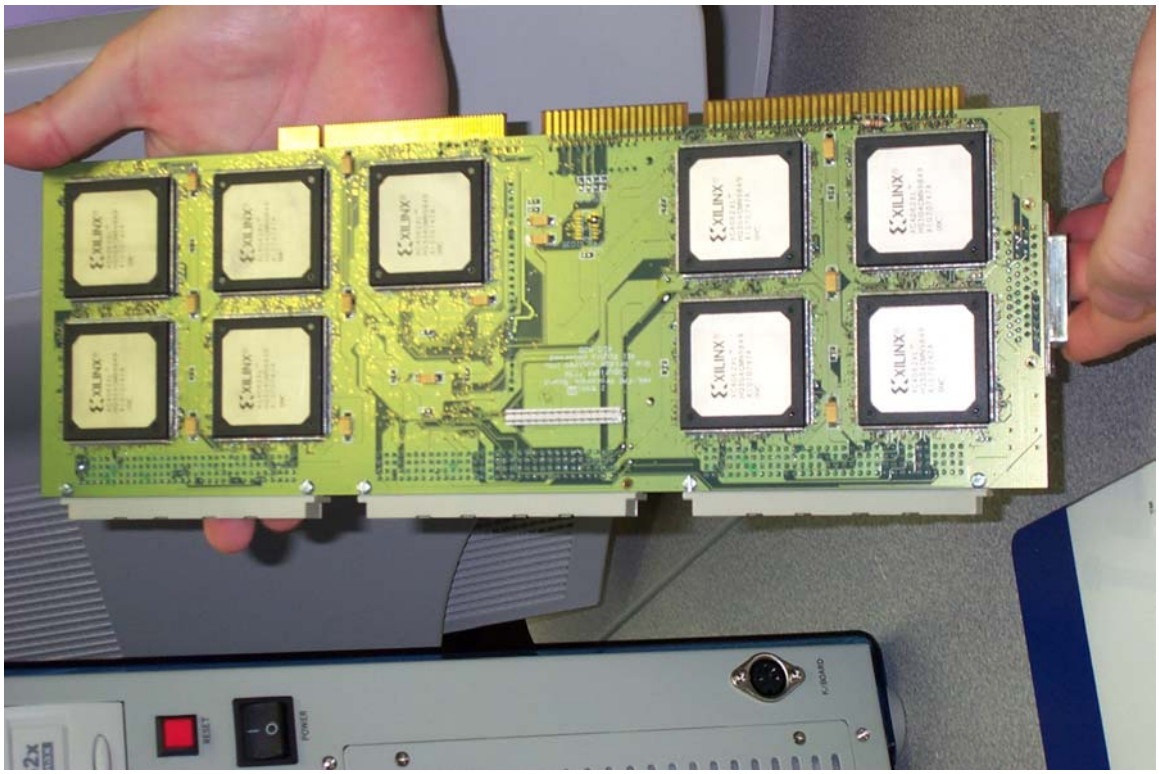


Figure 1: HAL-15 Recofigurable Computing Board with 10 Xilinx 6200 FPGA chips.

One or more FPGA boards are mounted in a chassis ,two of which are shown in Figure 2.

Figure 2: Two NASA HAL-15 Hypercomputers (each has Fig. 1 board with 10 FPGA chips)

The capability of Xilinx FPGAs used in current and future Star Bridge Systems Hypercomputers is growing rapidly (see Table 1). In this 18 month period, performance has grown from 4 to 470 GFLOPS, and future advances continue to be announced.

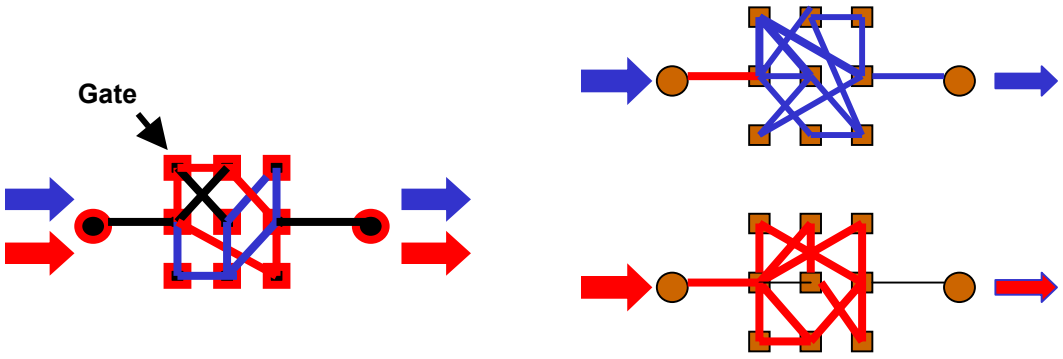
Xilinx FPGA	XC4062 (Feb '01)	XC2V6000 (Aug '02)
Gates	62,000	6,000,000 (97x)
Logic Cells	4608	67584 (12x)
Multiplies on FPGA	0	144
Clock Speed MHz	100	300 (3x)
Memory	20Kb	3.5 Mb (175x)
Memory Speed	466 Gb/s	5 Tb/s (11x)
Reconfigure Time	100ms	40ms (2.5x)
GFLOPS	0.4	47 (120x)
Total GFLOPS (10 FPGAs)	4	470

Table 1. Growth in FPGA Capability used in Star Bridge Hypercomputers

The 0.4 GFLOPS performance of the "older" XC4062 FPGA compares favorably with the 0.026 GFLOP performance of a Silicon Graphics Onyx MIPS10000 processor. More details are given in Applications. Unlike most FPGA applications that exploit only one FPGA, NASA algorithms have been tested which may exploit multiple FPGAs for floating point calculations. Some initial studies have even demonstrated extending calculations to two identical boards, each with 10 FPGAs.

Hypercomputer "Gateware"

Typical CPUs can perform multiple functions independently as shown in Figure 3. However, very little can be accomplished in parallel with CPUs as they process date primarily in a sequential manner with little opportunity for parallelism. Thus operations, such as the Blue and Red applications, may be accomplished, but with a large number of time steps.



Typical CPU FPGA

Figure 3. CPU and FPGA wired to perform 2 independent calculations (Blue & Red)

Traditionally, FPGAs are programmed as electronic circuits using 3 steps:
1. Very High-Level Design Language (VHDL) defining logic gate and connection layout
2. Translating VHDL to a circuit layout diagram to be traced and routed
3. Conversion to pages of esoteric ASCII code defining circuit details.

VIVA (Star Bridge's graphical language shown in Figure 4) reduces these steps to one step by replacing ASCII code with function graphical icons and connectors.

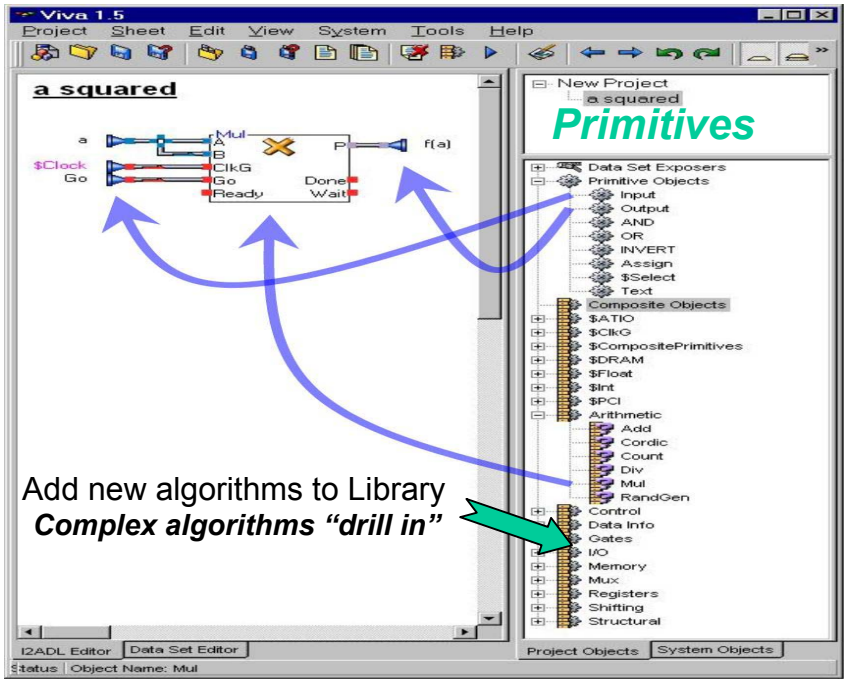


Figure 4. Coding "a-squared" function using Viva

The background of Figure 4 shows the Viva icons and drop-down menus at the top, primitive functions and library utilities on the right and system functions at the bottom. When beginning a new algorithm, the white working space on the left will be empty. To create an algorithm, such as "a squared, one picks the input horn icon, drags and drops it on the left of the white working area. Similarly an output icon horn is placed at the right of the working space. Then a multiply function icon is selected from the Arithmetic functions on the right and dragged and dropped between the input and output icons which are then connected, via transports to A and B inputs and P outputs, respectively. This is repeated for two other input horns. Function icon code may be displayed by clicking (drilling into) icons. Typically, one two-dimensional Viva page (with a 3rd dimension to drill in) is more understandable and often replaces many pages of traditional one-dimensional ASCII code. Table 2 shows the rapid advances in Viva capability.

VIVA 1.0 (Feb '01)	VIVA 2.0 (May '02)
NO Floating Point	Extensive data types
NO Scientific Functions	Trig, Logs, transcendentals
NO File Input/Output	File Input/Output
NO Vector-Matrix Support	Vector-Matrix Support
Access to One FPGA	Access to Multiple FPGAs
Primitive Documentation	Comprehensive Documentation/Training
Weekly Changes	Stable Development
Frequent "Bugs"	Few "Bugs"

Table 2: Growth in VIVA Capability

Application Algorithms

The first application shows the traditional programming of a do loop adding a+b from 1 to 1 billion on the left and a Viva algorithm for the same operation on the right of Figure 5.

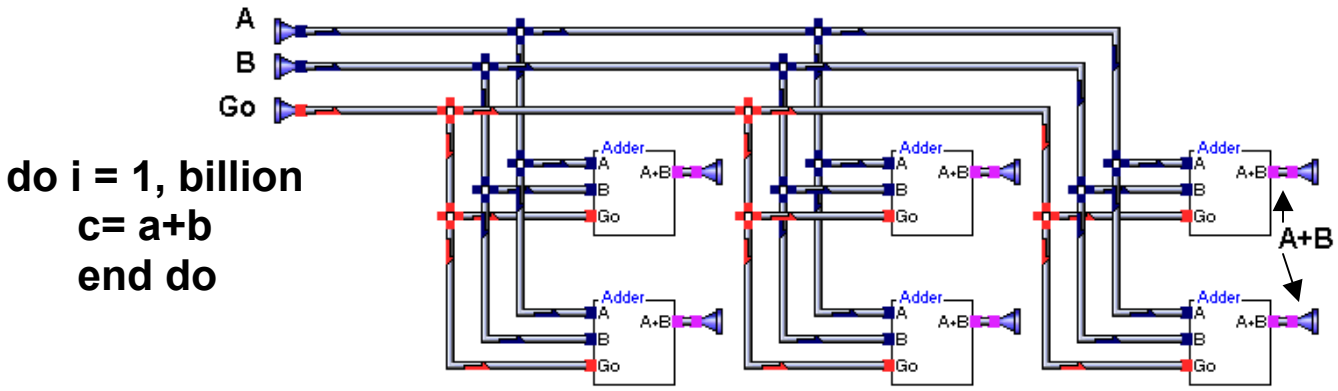


Figure 5. Traditional and Viva Algorithm to perform a+b 1 billion times

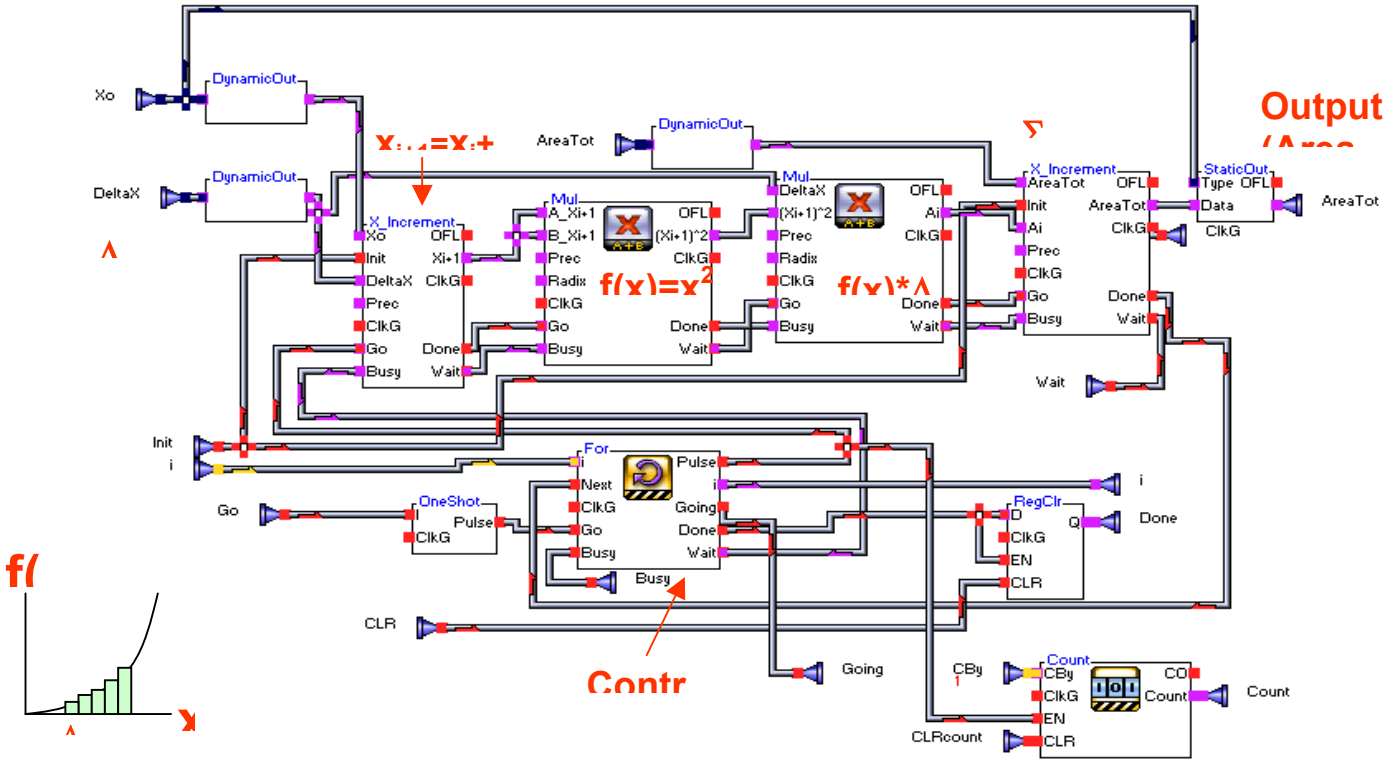


Figure 6. Viva Integration Algorithm (compute area under curve).

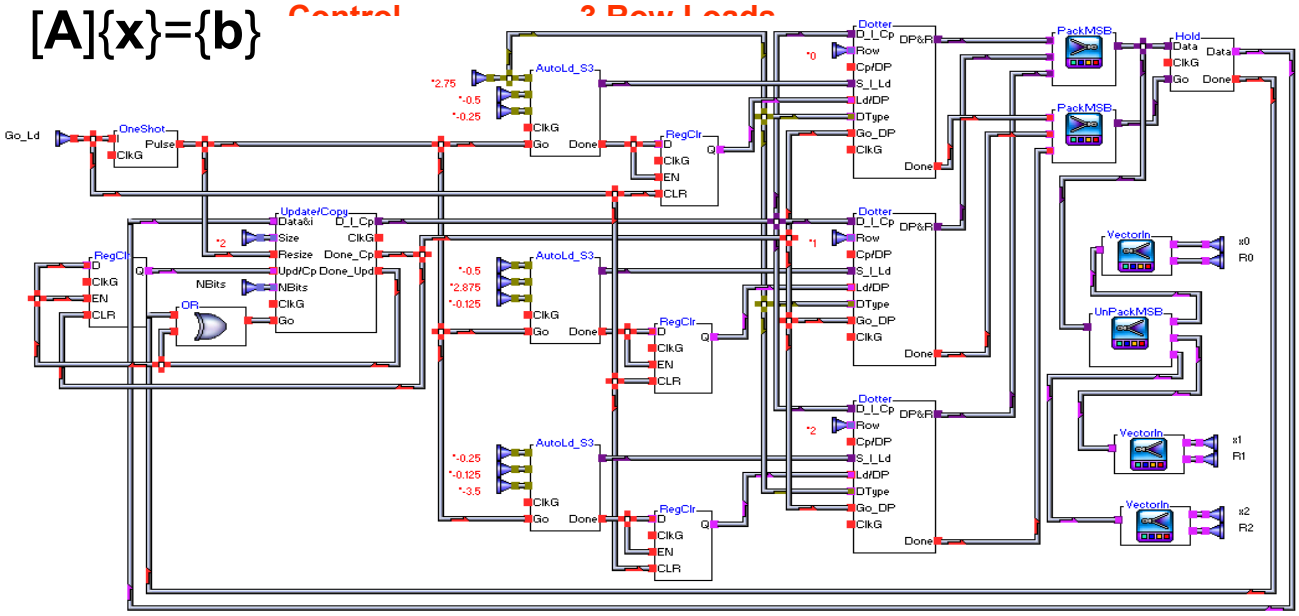


Figure 5. Viva Algorithm to solve linear system of matrix equations

NASA Langley, the first in the U.S. with a Star Bridge Hypercomputer, is collaborating with Star Bridge via weekly telecons to develop and extend VIVA. Despite initial VIVA 1.0 limitations, scientific and engineering applications were developed at NASA Langley to evaluate and exploit the Hypercomputer for matrix/vector operations, factorial, transcendental functions via Cordic algorithm (trig, log..), structural dynamics:

[M]x" + [C]x' + [K]x = f(t)

(where **[M], [C]** and **[K]** are the mass, damping and stiffness matrices, **f(t)** the force vector and **x** the displacement) and simultaneous matrix equation solution:

[K]{x} = {f}

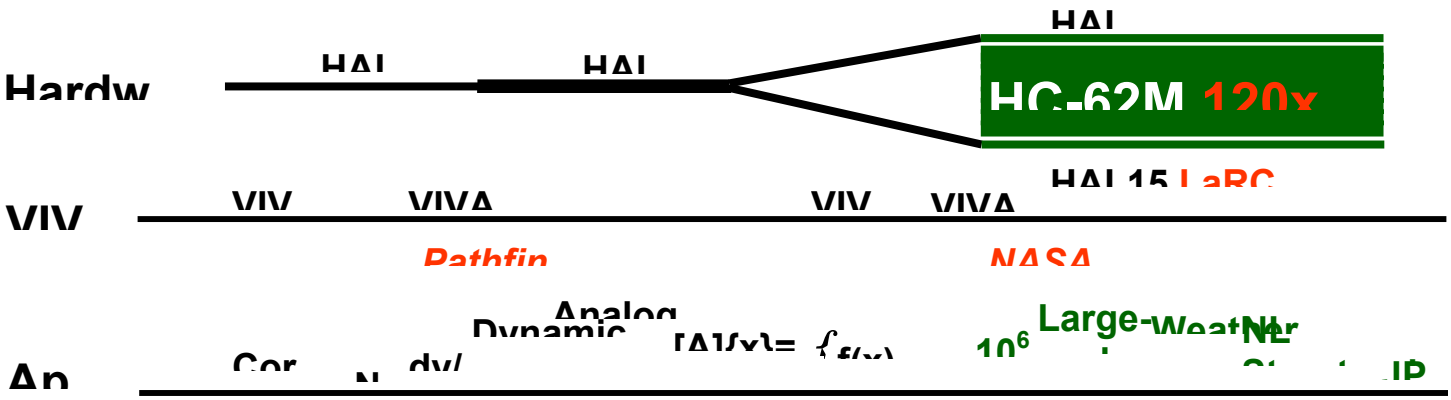
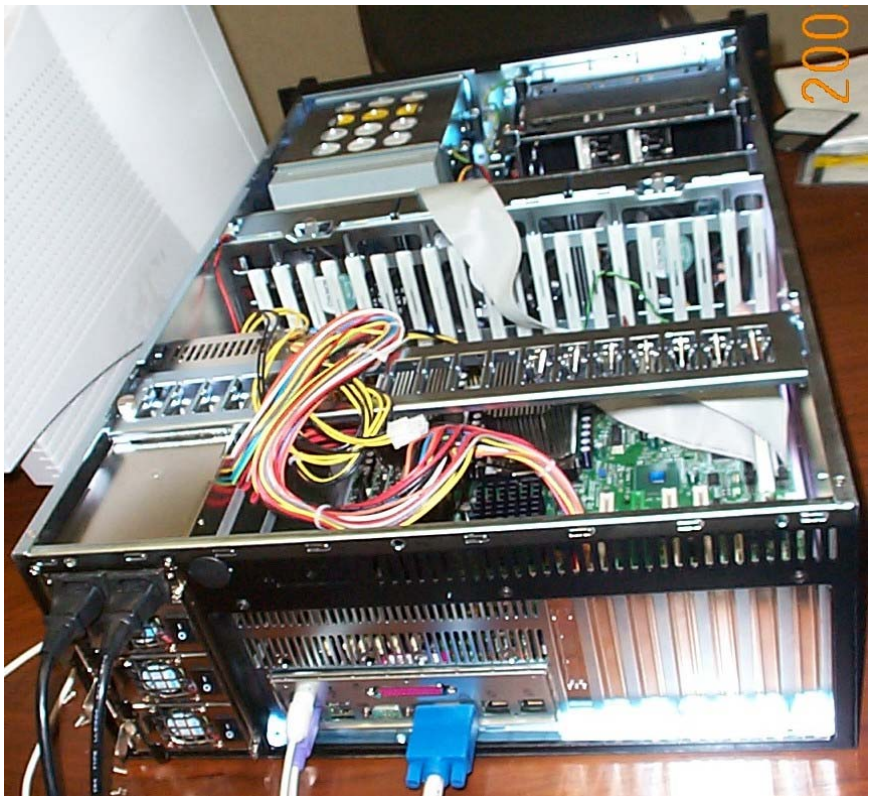


Figure 5. hardware, Viva and Application progress with time.



The paper describes algorithms enabling these applications and projects the impact of new gateware (using VIVA2.0 and HC-60M) to radically increase the size and range of new applications and significantly reduce their solution time.

References

1. Voss, David; Programmable Chips, MIT Technology Review, May 2002.
2. Singleterry, R., Sobieski, J. and Brown, S.; Field-Programmable Gate Array Computer in Structural Analysis: An Initial Exploration, AIAA 43rd SDM April 22-25, 2002.